

**NAVAL SURFACE WARFARE CENTER
PANAMA CITY DIVISION
PANAMA CITY, FLORIDA 32407-7001**



TECHNICAL REPORT
NSWC PCD TR 2015-003

OPTIMIZED WATERSPACE MANAGEMENT AND SCHEDULING USING MIXED-INTEGER LINEAR PROGRAMMING

Dr. Matthew J. Bays
Automation and Dynamics Branch
Unmanned Systems and Threat Analysis Division
Science and Technology Department

Dr. Thomas A. Wettergren
Code 00T1
Naval Undersea Warfare Center, Newport

January 2016

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE				<i>Form Approved OMB No. 0704-0188</i>	
<small>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</small>					
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)

Foreward

We describe an approach for accomplishing the high-level mission planning required for a heterogeneous team of autonomous vehicles performing mine countermeasure (MCM) survey missions in multiple areas. The high-level mission scheduling and waterspace management requires sequencing the order and location of lower-level MCM tasks to be completed by each vehicle in the heterogeneous team: unmanned surface vessels (USVs) and unmanned underwater vehicles (UUVs). We propose solving this complex sequencing operation by leveraging unique information processing, communication, refueling, and planning windows that form constraints within the system within a formal scheduling optimization framework known as mixed-integer linear programming. We pose the problem using a mixed-integer linear programming optimization framework, compare several complexity reduction heuristics to the full optimization, and include methods to account for relative uncertainty in the duration of planned tasks in such a manner to balance the risk of schedule slips or conflicts to the risk of creating an overly conservative and sub-optimal schedule.

This technical report has been prepared, reviewed, and approved by the Science and Technology Department.

BAYS.MATTHEW
J.1281374881

Digitally signed by
BAYS.MATTHEW.J.1281374881
DN: c=US, o=U.S. Government,
ou=DoD, ou=PKI, ou=USN,
cn=BAYS.MATTHEW.J.1281374881
Date: 2016.02.22 10:27:17 -06'00'

.....
Dr. Matthew J. Bays, Code X22,
Automation and Dynamics Branch

WETTERGREN.THOMAS.A.1230451849
MAS.A.1230451849

Digitally signed by
WETTERGREN.THOMAS.A.1230451849
DN: c=US, o=U.S. Government, ou=DoD,
ou=USN, ou=PKI, ou=USN,
cn=WETTERGREN.THOMAS.A.1230451849
Date: 2016.02.24 10:42:58 -05'00'

.....
Dr. Thomas A. Wettergren, Code 00T1,
Naval Undersea Warfare Center, Newport

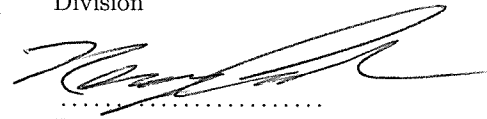
RODRIGUEZ.RAFAEL.1232170472
ELR.1232170472

Digitally signed by
RODRIGUEZ.RAFAEL.1232170472
DN: c=US, o=U.S. Government, ou=DoD, ou=PKI,
ou=USN, cn=RODRIGUEZ.RAFAEL.1232170472
Date: 2016.02.25 06:43:34 -06'00'

.....
Rafael Rodriguez, Code X22,
Automation and Dynamics Branch



.....
Daniel Kucik, Code X20,
Unmanned Systems and Threat Analysis
Division



.....
Dr. Kerry Commander, Code X,
Science and Technology Department

Contents

1	Introduction	4
1.1	Related Work	4
2	Mine Countermeasure Logistics Scheduling using MILP	5
3	Constraint Formulation	7
3.1	Docking and Deployment Constraints	7
3.2	Fuel Limitation and Charging Constraints	8
3.3	Goal and Capability Constraints	8
3.4	Time Dependency and Wait Constraints	8
3.5	Transit and Transport Constraints	10
3.6	Survey Constraints	10
3.7	Cost Function	11
3.8	Reacquire and Identify Scheduling Constraints	11
3.9	Neutralization Constraints	12
4	Complexity and Complexity Reduction	13
4.1	Node Reduction Strategy	15
4.2	Edge Reduction Strategy	15
5	Robust Scheduling Extensions	15
6	Simulation Results	19
6.1	End-to-End MCM Mission Scheduling	20
6.2	Deterministic and Robust Scheduling Results	20
6.3	Analysis of Complexity Reduction Heuristics and Computational Tractability	22
7	Conclusions	24
A	Distribution	27

List of Figures

1	Notional illustration of an MCM mission utilizing unmanned vehicles, including USVs to transport UUVs.	6
2	Comparison between full edge transit graph and graph from complexity reduction strategies.	16
3	Risk function for Schedule slip based on $\mathcal{A}_s(0, 20)$ with $c^{ineff} = T_{a,l}^{k,buff}$ and $c^{slip} = 200$	18
4	Plot of vehicle actions within an MCM area.	20
5	Simulated MCM Schedule.	21
6	Comparison between deterministic optimization, a naive robust optimization approach, and the proposed minimum-risk scheduling approach.	22
7	Histogram of Monte Carlo simulations showing minimum objective value found by different complexity reduction strategies.	23

List of Tables

1	Characterization of phase, agent count, and optimization type effects on optimization quality.	24
---	--	----

1 Introduction

The use of autonomous systems to perform increasingly complex and coordinated tasks has necessitated creating heterogeneous teams of agents, where different agent types specialize in different parts of an operation [1, 2, 3]. One such heterogeneous team operation is the heterogeneous unmanned system mine countermeasure (MCM) scenario, when one type of mobile agent, typically unmanned underwater vehicles (UUVs) is tasked with performing the direct surveys and re-acquire/identify tasks for the operation, while another type of larger, faster-moving, or longer-range agent, an unmanned surface vessel (USV) is responsible for transporting the UUVs between jobs for faster completion. Within manned systems, there are numerous examples of the USV/UUV concept such as aircraft carriers and their respective aircraft, garbage trucks and accompanying garbage workers, or mail delivery and their respective postmen. While this form of close interaction between unmanned systems is still far from common, the underlying hardware and guidance infrastructure to allow autonomous docking and deployment between unmanned systems is actively researched [4, 5, 6, 7, 8].

We present a high-level planner that takes the lower-level vehicle tasks required to complete a typical MCM survey mission and sequences them in a near-optimal manner to minimize MCM objective functions such as mission time while accomplishing MCM mission objectives. We use mixed-integer linear programming (MILP) as an optimization technique for our approach [9]. MILP has previously been used for probabilistic vehicle motion control, and is capable of quickly finding near-optimal solutions to constrained optimization problems involving linear and integer variables [10].

The principal theory developed in this research will be how to pose the MCM scheduling problem in a formal mathematical manner such that it can be input into an off-the-shelf MILP solver, while allowing for the inherent uncertainty in the system. MILP requires two fundamental components: a matrix of mathematical constraint equations that capture both the independent variables of optimization, as well as linear and binary variables that are dependent on those optimization variables. The second component is the linear cost function that is dependent on all independent and dependent variables used in the constraint equations. If these components are posed successfully to capture the dynamics of the MCM mission, solutions can be found relatively quickly. While the research will predominantly consider heterogeneous teams unmanned vehicles for the development of the optimization constraints, it is likely this research could be applied to teams of manned systems provided similar constraints on communication, refueling, or other mission characteristics exist.

1.1 Related Work

The problem of scheduling and task allocation has been extensively studied with a variety of techniques in multiple different forms. Koes et al developed a coordination architecture for modeling multirobot coordination and task allocation in [11]. Our framework complements this methodology by expanding the types of scenarios under which a constraint optimization framework can be used: namely to problems where there are strongly coupled constraints on vehicles such as transportation actions.

Korsah et al create a framework to explore optimal vehicle routing with cross-schedule constraints with applications to robotic assistance in [12]. However, there are several notable differences to our work. Firstly, the principal element of optimization is an agent *route*, not the collection of agent actions comprising the route. Additionally in Korsah's problem, the individuals being transported along the routes are static; there is no mechanism within the optimization framework for the individuals themselves to move about the area. As such [12] is similar to the traditional vehicle routing problem (VRP) [13]. Similarly, Mathew et al addresses finding the shortest route for a unmanned ground vehicle (UGV) and unmanned aerial vehicle (UAV) pair to transit and make deliveries to locations only reachable by the UAV in [14]. However, their work focuses primarily on the overall path planning for the two vehicles, and not on fuel limitations and the higher-level scheduling aspects required when there are an arbitrary number of UAVs and UGVs. Our work takes a scheduling-centric approach, formally incorporates fuel constraints, and is generalized for an arbitrary number of UUVs and USVs.

Gombolay et al develop a centralized algorithm to efficiently schedule manufacturing processes using robotic teams in [15]. However, they primarily focus on the development of a task sequencing heuristic

that allows the generalized problem, a Simple Temporal Problem [16], coupled with simple spacio-temporal constraints, to be tractable for large numbers of tasks and robots to be assigned to a fixed set of tasks. Our principal contribution is in the unique modeling and constraint-based logic required for the docking, transport, and deployment of unmanned vehicles throughout the environment. The development of the non-trivial constraints within our framework allows for applications where a list of required tasks is not given, but instead a variety of different methods are available to the vehicles to collectively complete a set of higher-level objectives by the vehicles' interaction.

Previous work specifically for MCM scheduling has typically focused on the task allocation problem [17, 1]. For example, several efforts have been undertaken to determine optimal partitioning of search areas for undersea search with various assumptions on sensor modeling and false alarm information in [18, 19]. Our work extends these results by focusing on the *when* and *how* to optimally have vehicles prosecute different mission areas, as opposed to simply *where* vehicles should be tasked. More recently, work by Molieneaux et al have developed algorithms to aid in schedule repair for previously-developed MCM operation schedules using a cognitive architecture framework [20]. With their framework, the authors attempt to repair user-designed MCM schedules that have become infeasible due to unforeseen circumstances such as asset breakdown or tasking delays. In our work, we attempt to solve the initial step by providing a formal, automated method to plan an overall MCM schedule from a mathematical model of generalized MCM operation constraints.

General robust scheduling techniques have been studied using a variety of methods. Lin, Janak, and Floudas develop complimentary approaches to formally incorporate uncertainty in scheduling problems in [21] and [22]. In their work, Lin et al develop formal methodologies for converting MILP scheduling problems where model parameters are uncertain under either a bound or known probability distribution into another MILP or MINLP-type problem where the uncertainty is explicitly incorporated into the optimization framework, yielding a robust solution that still meets the problem constraints with a fixed probability. In a similar manner, Bertsimas and Sim create a method to adjust the conservatism of robust MILP solutions in [23] while keeping the robust problem linear. Other work on robust scheduling and integer programming problems in the general sense may be found in [24, 25, 26]. However, to our knowledge there is no work explicitly incorporating Bayes risk to schedule slips nor for the application of MCM scheduling. Our work has the similar goal for the specific problem of MCM scheduling, but instead of robustly optimizing with a fixed uncertainty of meeting time constraints or an acceptable number of constraints that may be violated, we incorporate a cost function derived from Bayes risk using an adjustable cost of a schedule slip and the potential increase in buffer time for a task.

This paper is outlined as follows: in Section 2, we discuss the general setup of the the MCM scheduling problem, including the definition of the theoretical base problem we call the service agent transport problem (SATP). In Section 3, we outline the mathematical constraints that make up the framework. We propose several heuristics in order to limit the computational complexity in 4. We discuss robust scheduling extensions that utilize a cost function based on Bayes risk which adds buffer times in between tasks into the system in Section 5. Finally, simulation results showing the performance of various implementations of the SATP are discussed in Section 6.

2 Mine Countermeasure Logistics Scheduling using MILP

We will now outline an MCM scheduling problem by developing a formal mathematical framework with which to ultimately optimize a vehicle schedule. Figure 1 shows a notional illustration of the survey UUV transport problem. Let there exist a set of autonomous vehicles $\mathcal{A} = \{1, \dots, A\}$ that are tasked with surveying a number of survey areas $\mathcal{S} = \{1, \dots, S\}$. The survey areas are connected by a directed graph $\langle \mathcal{D}, \mathcal{V} \rangle$. The nodes $\mathcal{D} = \{o, 1, \dots, D\}$ represent locations at which an agent $a \in \mathcal{A}$ may enter a survey area $s \in \mathcal{S}$ in order to perform operations. The node labeled o is the origin node, which represents the starting point for all valid paths in the graph. The agents may move from node i to node j along an edge of the graph $v = (i, j) \in \mathcal{V}$. The subset $\mathcal{D}_s \subset \mathcal{D}$ represents nodes from which agents may survey area s .

The use of graph representations of scheduling problems has been historically used effectively due to the ability of the methods to directly lead to effective heuristic decomposition procedures [27] that can greatly reduce the computational complexity of the optimization. Structural simplifications of the graph structure may be used to create heuristics on the survey scheduling; for example, the resulting optimized schedule is

expected to be from the subset of acyclic paths within the graph structure. Thus, reducing the problem size by decomposing the graph structure into a directed acyclic graph structure is a reasonable complexity reduction strategy; such an approach is considered in Section 4.

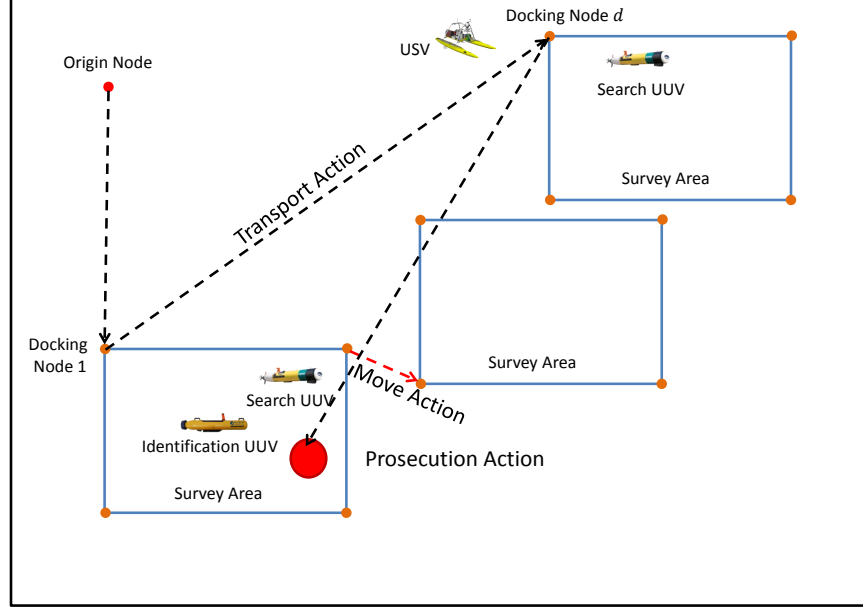


Figure 1: Notional illustration of an MCM mission utilizing unmanned vehicles, including USVs to transport UUVs.

The vehicles are separated into three mutually exclusive and collectively exhaustive subsets \mathcal{A}_{ur} , \mathcal{A}_r , and \mathcal{A}_s consisting of survey UUVs, reacquire and identify (RI) UUVs, and USVs, respectively. A survey UUV $m \in \mathcal{A}_{ur}$ is capable of directly surveying the survey areas. An RI UUV $r \in \mathcal{A}_r$ perform RI actions on areas previously surveyed and in which MILCOS have been found. A USV $n \in \mathcal{A}_s$ may collect a fixed number of the UUVs and transport them between nodes. For some of the below constraints, we also wish to define the union of survey UUVs and RI UUVs as $\mathcal{A}_{ur} = \mathcal{A}_u \cup \mathcal{A}_r$. The UUVs may move between nodes with a time cost c_{ij}^{move} , however are significantly slower than USVs with transportation cost of $c_{ij}^{trans} < c_{ij}^{move}$. The USVs also require a fixed amount of time to collect and deploy the survey UUVs. The collection and deployment costs are denoted c^{dock} and c^{deploy} , respectively. The vehicles initially start at an origin node o , with each USV n containing A_n UUVs.

We assume it is known due to enemy course of action (ECOA) intelligence, environmental information, and other sources that there is an expected number of mine-like contacts (MILCOs) \hat{N}_s^{MILCO} before completion of survey actions within area s , as well as an expected number of mines \hat{N}_s^{mines} before completion of RI actions in area s . Upon completion of the survey and RI actions, these quantities and their expected locations will be known, and denoted as N_s^{MILCO} and N_s^{mines} , respectively. Until the locations are known from the actual survey and RI actions, we assume the location of the MILCOs and mines follow density functions $p_{MILCO}(s)$ and $p_{mine}(s)$. Using the expected number of MILCOs and mines, we assume that expected time costs for performing RI and neutralize actions can be estimated for each survey area s . We denote the expected RI and neutralization costs as $\hat{c}_{r,s}^{RI}$ and $\hat{c}_{n,s}^{neut}$ using vehicle r or n on survey area s .

We outline the general logistical constraints for an operation where multiple UUVs may be transported throughout the field via an USV based largely on [28]. We then expand the logistical constraints with the specific constraints developed for general survey, reacquire and identify (RI), and prosecution tasking such as found in MCM operations.

The primary optimization variables for our specific solution to the SATP, which we call *decision variables*,

fall into two groups: scheduling indicator variables and time variables. The scheduling indicator variable $\bar{\mathbf{I}}_{a,p,\mathcal{L}}^k \in \{0,1\}$ denotes that task type k is scheduled for completion by vehicle a during phase p at location set $l \in \mathcal{L}_k$, where \mathcal{L}_k is the set of all feasible locations required for execution of task type k and $\bigcup_{k \in \mathcal{K}} \mathcal{L}_k = \mathcal{L}$. The timing variables $\bar{\mathbf{T}}_{a,p}^{start} \in \mathbb{R}$ and $\bar{\mathbf{T}}_{a,p}^{end} \in \mathbb{R}$ denote the times at which the task of vehicle a during phase p is scheduled to start and end, respectively. In contrast to the decision variables, additional variables are needed for the optimization problem in order to develop the necessary constraints to formulate the optimization problem. While the additional variables, which we denote as *helper variables* are technically variables used in the optimization, they are not of primary concern for the optimization problem. We will discuss the helper variables as the details of the optimization problem are formulated.

3 Constraint Formulation

We now turn to defining the constraints required for the mathematical formulation of the MCM scheduling problem pertaining to the survey constraints and logistics management. The following constraints were originally characterized within the SATP in [28].

3.1 Docking and Deployment Constraints

The first constraint within the SATP is that a UUV must only complete certain tasks when they are docked or deployed, as appropriate. For enforcing this constraint, we create the helper variables to track whether a survey UUV m is docked or deployed from a USV n . We denote this variable

$$\bar{\mathbf{D}}_{mn,p} = \begin{cases} 1 & \text{UUV } m \text{ is docked with USV } n \text{ in phase } p \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Within an integer programming framework, we create (1) with the formal constraint

$$\begin{aligned} \forall m \in \mathcal{A}_{ur}, \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P} \\ \bar{\mathbf{D}}_{mn,p} = \sum_{p' \leq p, d \in D} \left[\bar{\mathbf{I}}_{mn,p',d}^{dock} - \bar{\mathbf{I}}_{mn,p',d}^{deploy} \right], \end{aligned} \quad (2)$$

where $\bar{\mathbf{I}}_{mn,p',d}^{dock}$ and $\bar{\mathbf{I}}_{mn,p',d}^{deploy}$ are indicator variable docking and deploying, respectively, at particular dock/deploy nodes d . We bound $\bar{\mathbf{D}}_{mn,p}$ with the constraint

$$\forall m \in \mathcal{A}_{ur}, \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P} \quad 0 \leq \bar{\mathbf{D}}_{mn,p} \leq 1. \quad (3)$$

With constraints (2) and (3), $\bar{\mathbf{D}}_{mn,p}$ equals 1 if UUV m is currently docked with USV n during phase p , and 0 otherwise. Constraint (2) and the lower bound in (3) satisfies the additional requirement that in order for a survey UUV to be deployed, it must first have been docked. It is also necessary to track if an UUV is docked or deployed in general. This is performed using the helper variable $\bar{\mathbf{D}}_{mn,p}$ and constraints

$$\forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad \bar{\mathbf{D}}_{m,p} = \sum_{n \in \mathcal{A}_s} \bar{\mathbf{D}}_{mn,p} \quad (4)$$

and

$$\forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad 0 \leq \bar{\mathbf{D}}_{m,p} \leq 1 \quad (5)$$

$$(6)$$

Additionally, we limit the total number of UUVs that can be docked at a given time by a USV to D_{max} using the constraint

$$\forall n \in \mathcal{A}_s, \forall p \in \mathcal{P} \quad \sum_{m \in \mathcal{A}_{ur}} \bar{\mathbf{D}}_{mn,p} \leq D_{max} \quad (7)$$

With constraints (4)-(7), we can now limit the feasible actions of the individual vehicles when deployed or docked as appropriate.

3.2 Fuel Limitation and Charging Constraints

Fuel consumption is an additional variable that must be tracked for each vehicle during all phases of the mission. For tracking fuel, we define the variable $\bar{\mathbf{F}}_{m,p} \in \mathbb{R}$ as the amount of fuel survey UUV m has available during the start of phase P . The variable $\bar{\mathbf{F}}_{m,p}$ is assigned a value using the constraints

$$\begin{aligned} \forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad (8) \\ \bar{\mathbf{F}}_{m,p} = \sum_{p' \leq p} \left[\bar{\mathbf{F}}_{m,p'}^{charge} - \sum_{s \in \mathcal{S}} f_{m,s}^{survey} \bar{\mathbf{I}}_{m,p',s}^{survey} \right. \\ \left. - \sum_{d \in \mathcal{D}} \left(f_{m,d}^{dock} \bar{\mathbf{I}}_{m,p',d}^{dock} + f_{m,d}^{deploy} \bar{\mathbf{I}}_{m,p',d}^{deploy} \right) \right. \\ \left. - \sum_{v \in \mathcal{V}} f_{m,p',v}^{move} \bar{\mathbf{I}}_{m,p',v}^{move} \right] \end{aligned}$$

where $f_{m,s}^{survey}$, $f_{m,d}^{dock}$, $f_{m,d}^{deploy}$, and $f_{m,v}^{move}$ are fixed costs for executing their respective goals, and $\bar{\mathbf{F}}_{m,p}^{charge}$ is a floating variable indicating the amount a vehicle is charged during a phase. The variable $\bar{\mathbf{F}}_{m,p}^{charge}$ is fixed to its value using the linear survey UUV charge rate c_m^{charge} and the constraints

$$\begin{aligned} \forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad (9) \\ \bar{\mathbf{D}}_{m,p} = 1 \Rightarrow \bar{\mathbf{F}}_{m,p}^{charge} \leq c_m^{charge} (\bar{\mathbf{T}}_{m,p}^{end} - \bar{\mathbf{T}}_{m,p}^{start}) \\ \bar{\mathbf{D}}_{m,p} = 0 \Rightarrow \bar{\mathbf{F}}_{m,p}^{charge} = 0. \end{aligned}$$

Additionally, we limit the fuel capacity with the constraints

$$\forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad (10)$$

$$\bar{\mathbf{F}}_{m,p} \leq F_{max}$$

$$\forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad (11)$$

$$0 < \bar{\mathbf{F}}_{m,p}$$

3.3 Goal and Capability Constraints

In our formulation, each vehicle can only prosecute one goal during a phase of the mission. As such, we have the constraint

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P} \quad \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \bar{\mathbf{I}}_{a,p,l}^k \leq 1. \quad (12)$$

As not every vehicle is capable of completing every task, this requires additional constraints to force only vehicles capable of completing a task to be scheduled for the task. As such, we have the constraint

$$\forall a \notin \mathcal{A}_k, \forall p \in \mathcal{P}, \forall l \in \mathcal{L}_k \quad \bar{\mathbf{I}}_{a,p,l}^k = 0, \quad (13)$$

where \mathcal{A}_k is the set of vehicles that are capable of action k . We note that this constraint may also be implemented for tractability by limiting both the scheduling variables $\bar{\mathbf{I}}_{a,p,l}^k$ and constraints specific to a certain capability that are created in the optimization problem to those allowed by the individual vehicle's capabilities.

3.4 Time Dependency and Wait Constraints

We now turn to the constraints required for determining the specific times at which phases are scheduled for starting and ending. Without loss of generality, we assume that the start time of the initial phase is at $t = 0$. As such we have the constraint

$$\forall a \in \mathcal{A} \quad \bar{\mathbf{T}}_{a,p=0}^{start} = 0. \quad (14)$$

In order to adhere to the boundary conditions between the end of the last phase and the start of the next phase, we have the constraint

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P} \mid p \neq 0 \quad \bar{\mathbf{T}}_{a,p+1}^{start} \geq \bar{\mathbf{T}}_{a,p}^{end} \quad (15)$$

With constraints (14) and (15), we have defined all variables with the exception of the end time of each phase, $\bar{\mathbf{T}}_{a,p}^{end}$. The end time may be defined by taking advantage of (12) with only one task being scheduled for a vehicle during a phase

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P} \quad \bar{\mathbf{T}}_{a,p}^{end} = \bar{\mathbf{T}}_{a,p}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} c_{a,l}^k \bar{\mathbf{I}}_{a,p,l}^k \quad (16)$$

where $c_{a,L}^k$ is the duration of task k if performed by vehicle a on location set L . Additionally, the dual relationship between a UUV and USV performing docking and deployment constraints necessitates that we must ensure that those particular operations occur at the same time for both vehicles. Thus, we have the time dependency constraints

$$\forall m \in \mathcal{A}_{ur}, \forall m \in \mathcal{A}_s, \forall p \in \mathcal{P} \quad (17)$$

$$\sum_{d \in \mathcal{D}} \bar{\mathbf{I}}_{mn,p,d}^{dock} = 1 \Rightarrow \bar{\mathbf{T}}_{m,p}^{start} = \bar{\mathbf{T}}_{n,p}^{start}$$

$$\forall m \in \mathcal{A}_{ur}, \forall m \in \mathcal{A}_s, \forall p \in \mathcal{P} \quad (18)$$

$$\sum_{d \in \mathcal{D}} \bar{\mathbf{I}}_{mn,p,d}^{deploy} = 1 \Rightarrow \bar{\mathbf{T}}_{m,p}^{end} = \bar{\mathbf{T}}_{n,p}^{end}$$

Some scheduling situations necessitate an vehicle loitering or otherwise performing no meaningful actions during a given phase so that it may properly coordinate with the actions of other vehicles. For example, a USV may need to wait at a docking point during a phase in which a survey UUV is servicing an area with which it will dock in the proceeding phase. In this case, we require an additional variable indicating that a vehicle is idle during that phase at a given location. We define the $\bar{\mathbf{I}}_{a,p,d}^{wait}$ indicator variable by the constraints

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P}, \forall d \in \mathcal{D} \quad (19)$$

$$\sum_{k \in \mathcal{K}} \left[\sum_{l \in \mathcal{L}_k} \bar{\mathbf{I}}_{a,p,l}^k \right] + \sum_{d' \in \mathcal{D} \mid d' \neq d} \bar{\mathbf{I}}_{a,p,d'}^{wait} = 1 \Rightarrow \bar{\mathbf{I}}_{a,p,d}^{wait} = 0$$

$$\forall a \in \mathcal{A}, \forall k \in \mathcal{K}, \forall p \in \mathcal{P}, \forall d \in \mathcal{D} \quad (20)$$

$$\sum_{k \in \mathcal{K}} \left[\sum_{\substack{l \in \mathcal{L}_k \\ d \notin \mathcal{L}}} \bar{\mathbf{I}}_{a,p-1,l}^k \right] + \sum_{\substack{d' \in \mathcal{D} \\ d' \neq d}} \bar{\mathbf{I}}_{a,p-1,d'}^{wait} = 1 \Rightarrow \bar{\mathbf{I}}_{a,p,d}^{wait} = 0$$

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \bar{\mathbf{I}}_{a,p,l}^k = 0 \Rightarrow \sum_{d \in \mathcal{D}} \bar{\mathbf{I}}_{a,p,d}^{wait} = 1. \quad (21)$$

Constraint (19) forces $\bar{\mathbf{I}}_{a,p,d}^{wait}$ to equal zero if the vehicle is otherwise tasked or waiting at another location. Constraint (20) forces $\bar{\mathbf{I}}_{a,p,d}^{wait}$ to equal zero if the vehicle performed an action at a different location d' during the previous time step, thereby prohibiting the vehicle from waiting at location d . Constraint (21) forces $\bar{\mathbf{I}}_{a,p,d}^{wait}$ to one if the vehicle is not otherwise tasked.

3.5 Transit and Transport Constraints

We now turn to the constraints required for transiting vehicles and transporting them. In order for a vehicle to transit from one location to the next, it must have performed an action at the previous location

$$\begin{aligned}
& \forall m \in \mathcal{A}_{ur}, \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d_0 \in D \\
& \sum_{d \in \mathcal{D}} \bar{\mathbf{I}}_{m,p,(d_0,d)}^{move} = 1 \\
& \Rightarrow \sum_{d' \in \mathcal{D}} \bar{\mathbf{I}}_{m,p-1,(d',d_0)}^{move} + \sum_{n \in N} \bar{\mathbf{I}}_{mn,p-1,d_0}^{deploy} \\
& + \sum_{s \in \mathcal{S} | d_0 \in \mathcal{D}_s} \bar{\mathbf{I}}_{m,p-1,s}^{survey} + \bar{\mathbf{I}}_{m,p-1,d}^{wait} = 1.
\end{aligned} \tag{22}$$

where $(d_0, d), (d', d_0) \in \mathcal{V}$. Additionally, we have the constraint that for a vehicle to either move from one area to the next or to survey an area, the vehicle must be deployed

$$\forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad \sum_{v \in \mathcal{V}} \bar{\mathbf{I}}_{m,p,v}^{move} = 1 \Rightarrow \bar{\mathbf{D}}_{m,p} = 0 \tag{23}$$

For a UUV m to be docked with USV n , both vehicles must have been located previously at the location of the capture. Thus, we have

$$\forall m \in \mathcal{A}_{ur}, \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d \in \mathcal{D} \tag{24}$$

$$\begin{aligned}
& \bar{\mathbf{I}}_{mn,p,d}^{dock} = 1 \Rightarrow \sum_{d_{01} \in D} \bar{\mathbf{I}}_{m,p-1,(d_{01},d)}^{move} + \bar{\mathbf{I}}_{m,p-1,d}^{wait} \\
& + \sum_{s \in \mathcal{S} | d \in \mathcal{D}_s} \bar{\mathbf{I}}_{m,p-1,s}^{survey} = 1 \\
& \bar{\mathbf{I}}_{mn,p,d}^{dock} = 1 \Rightarrow \sum_{d_{01} \in D} \bar{\mathbf{I}}_{n,p-1,(d_{01},d)}^{move} \\
& + \bar{\mathbf{I}}_{n,p-1,d}^{wait} \\
& + \sum_{\substack{m' \in \mathcal{A}_{ur} \\ m' \neq m}} \left[\bar{\mathbf{I}}_{m'n,p-1,d}^{dock} + \bar{\mathbf{I}}_{m'n,p-1,d}^{deploy} \right] \geq 1
\end{aligned} \tag{25}$$

where $(d_{01}, d), (d_{02}, d) \in \mathcal{V}$. Likewise, we have the constraint for deploying UUVs as

$$\forall m \in \mathcal{A}_{ur}, \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d \in \mathcal{D} \tag{26}$$

$$\begin{aligned}
& \bar{\mathbf{I}}_{mn,p,d}^{deploy} = 1 \Rightarrow \bar{\mathbf{D}}_{mn,p} = 1 \\
& \bar{\mathbf{I}}_{mn,p,d}^{deploy} = 1 \Rightarrow \sum_{d_{01} \in D} \bar{\mathbf{I}}_{mn,p-1,(d_{01},d)}^{move} \\
& + \bar{\mathbf{I}}_{n,p-1,d}^{wait} + \sum_{\substack{m' \in \mathcal{A}_{ur} \\ m' \neq m}} \left[\bar{\mathbf{I}}_{m'n,p-1,d}^{dock} + \bar{\mathbf{I}}_{m'n,p-1,d}^{deploy} \right] \geq 1
\end{aligned} \tag{27}$$

3.6 Survey Constraints

The first constraint for a servicing action is that a survey area can only be surveyed if it is scheduled after a deployment action at a node $d \in \mathcal{D}_s$ or move action. Thus, we have the constraint

$$\forall m \in \mathcal{A}_u, \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d \in \mathcal{D}, \forall s \in \mathcal{S} \tag{28}$$

$$\bar{\mathbf{I}}_{m,p,s}^{survey} = 1 \Rightarrow \sum_{n \in \mathcal{A}_s} \bar{\mathbf{I}}_{mn,p-1,d}^{deploy} + \sum_{d' \in D} \bar{\mathbf{I}}_{m,p-1,(d',d)}^{move} = 1.$$

where $(d', d) \in \mathcal{V}$ Next, we have the constraint that a survey action can only be performed if it is scheduled while the UUV is deployed

$$\forall m \in \mathcal{A}_{ur}, \forall p \in \mathcal{P} \quad \sum_{s \in \mathcal{S}} \bar{\mathbf{I}}_{m,p,s}^{survey} = 1 \Rightarrow \bar{\mathbf{D}}_{m,p} = 0 \quad (29)$$

Additionally, every survey goal should be completed exactly once by the group of vehicles. Thus,

$$\forall s \in \mathcal{S} \quad \sum_{m \in \mathcal{A}_u} \sum_{p \in \mathcal{P}} \bar{\mathbf{I}}_{m,p,s}^{survey} = 1. \quad (30)$$

3.7 Cost Function

The cost function to be minimized in our initial formulation is to minimize the end time of the last phase over all vehicles. That is, our objective is

$$\text{minimize } \max_a \bar{\mathbf{T}}_{a,P}^{end}. \quad (31)$$

3.8 Reacquire and Identify Scheduling Constraints

An example of such mixed-integer linear program constraints is shown in the following. Let $\{p, p', p''\}$ represent indices in the set of mission phases \mathcal{P} . Index s iterates over the set \mathcal{S} of survey areas, and $d \in \mathcal{D}$ represents the set of dock/deploy points for the mission where UUVs may onboard or offboard the USV. Three criteria must be met prior to the RI action:

1. the area s must have been surveyed by a survey UUV,
2. the survey UUV m from survey UUV set \mathcal{A}_u must have been docked with a USV,
3. sufficient time must have elapsed from when the survey UUV had docked with a USV to perform post-mission analysis (PMA).

The first two criteria may be captured by the constraint

$$\begin{aligned} \forall r \in \mathcal{A}_r, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \quad (32) \\ \bar{\mathbf{I}}_{r,p,s}^{RI} = 1 \Rightarrow \sum_{m \in \mathcal{A}_{ur}} \left[\sum_{p' < p} \left[\sum_{n \in \mathcal{A}_s, d \in \mathcal{D}} \bar{\mathbf{I}}_{m,p',s}^{dock} \geq 1 \right. \right. \\ \left. \left. \wedge \sum_{p'' < p'} \bar{\mathbf{I}}_{m,p'',s}^{survey} \geq 1 \right] \right] \geq 1, \end{aligned}$$

where $\bar{\mathbf{I}}_{r,p,s}^{RI}$ is the action indicator variable for scheduling RI UUV r performing an RI action during phase p in area s . In order to capture the third criteria, we create the constraint

$$\begin{aligned} \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d_n \in \mathcal{D}_n \quad (33) \\ \bar{\mathbf{I}}_{n,p,d_n}^{RI} = 1 \Rightarrow \bar{\mathbf{T}}_{n,p}^{start} \geq \bar{\mathbf{T}}_s^{PMA,end}, \end{aligned}$$

where $\bar{\mathbf{T}}_{m,s}^{PMA,end}$ is a new time indicator variable indicating the time PMA has ended for the RI component of an operation on survey area s , which is defined using the constraints

$$\begin{aligned}
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\
& \sum_{m \in \mathcal{A}_{ur}} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}_{m,p',s}^{survey} \geq 1 \right] \wedge [\bar{\mathbf{I}}_{m,p,s}^{dock} \geq 1] \right] \geq 1 \Rightarrow \\
& \bar{\mathbf{T}}_{n,p,s}^{PMA,start} = \bar{\mathbf{T}}_{n,p}^{end} \\
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\
& \sum_{M \in \mathcal{A}_{ur}} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}_{m,p',s}^{survey} \geq 1 \right] \wedge [\bar{\mathbf{I}}_{m,p,s}^{dock} \geq 1] \right] = 0 \Rightarrow \\
& \bar{\mathbf{T}}_s^{PMA,start} = \bar{\mathbf{T}}^{big} \\
& \forall s \in \mathcal{S} \\
& T_s^{PMA,start} = \min_{n \in \mathcal{A}^{trans}, p \in \mathcal{P}} T_{n,p,s}^{PMA,start} \\
& \forall s \in \mathcal{S} \\
& \bar{\mathbf{T}}_s^{PMA,end} = \bar{\mathbf{T}}_s^{PMA,start} + \hat{c}_s^{PMA},
\end{aligned} \tag{34}$$

where $\bar{\mathbf{T}}^{big}$ is a large constant.

3.9 Neutralization Constraints

We now turn to determining a proper schedule for the neutralization phase of a mission. In the scheduling of neutralization activities of a given survey area s , three criteria must be met prior to the Neutralization action:

1. the reacquire and identification task must have been performed on area s by an RI UUV $r \in \mathcal{A}_r$,
2. the RI UUV r must have been docked with a USV,
3. sufficient time must have elapsed from when the survey UUV had docked with a USV to perform post-mission analysis.

The first two criteria may be captured by the constraint

$$\begin{aligned}
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d_n \in \mathcal{D}_n \\
& \bar{\mathbf{I}}_{n,p,d_n}^{neut} = 1 \Rightarrow \sum_{r \in \mathcal{A}_r} \left[\sum_{p' < p} \left[\sum_{n \in \mathcal{A}_s} \bar{\mathbf{I}}_{rn,p',s}^{dock} \geq 1 \right. \right. \\
& \left. \left. \wedge \left[\sum_{p'' < p'} \bar{\mathbf{I}}_{r,p'',s}^{RI} \geq 1 \right] \right] \right] \geq 1,
\end{aligned} \tag{35}$$

where $\bar{\mathbf{I}}_{n,p,d_n}^{neut}$ is the action indicator variable for scheduling USV n to perform an neutralization action during phase p at the neutralization point $d_n \in \mathcal{D}_n$. In order to capture the third criteria, we create the constraint

$$\begin{aligned}
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall d_n \in \mathcal{D}_n \\
& \bar{\mathbf{I}}_{n,p,d_n}^{neut} = 1 \Rightarrow \bar{\mathbf{T}}_{n,p}^{start} \geq \bar{\mathbf{T}}_s^{PMA',end},
\end{aligned} \tag{36}$$

where $\bar{\mathbf{T}}_{m,s}^{PMA',end}$ is a new time indicator variable indicating the time PMA has ended for the RI component of an operation on survey area s , which is defined using the constraints

$$\begin{aligned}
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\
& \sum_{r \in \mathcal{A}_r} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}_{r,p',s}^{RI} \geq 1 \right] \wedge [\bar{\mathbf{I}}_{r,p,s}^{dock} \geq 1] \right] \geq 1 \Rightarrow \\
& \bar{\mathbf{T}}_{n,p,s}^{PMA',start} = \bar{\mathbf{T}}_{n,p}^{end} \\
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P}, \forall s \in \mathcal{S} \\
& \sum_{r \in \mathcal{A}_r} \left[\left[\sum_{p' \leq p} \bar{\mathbf{I}}_{r,p',s}^{RI} \geq 1 \right] \wedge [\bar{\mathbf{I}}_{r,p,s}^{dock} \geq 1] \right] = 0 \Rightarrow \\
& \bar{\mathbf{T}}_s^{PMA',start} = \bar{\mathbf{T}}^{big} \\
& \forall s \in \mathcal{S} \\
& T_s^{PMA',start} = \min_{n \in \mathcal{A}^{trans}, p \in \mathcal{P}} T_{n,p,s}^{PMA',start} \\
& \forall s \in \mathcal{S} \\
& \bar{\mathbf{T}}_s^{PMA',end} = \bar{\mathbf{T}}_s^{PMA',start} + \hat{c}_s^{PMA'}.
\end{aligned} \tag{37}$$

With the above constraints, we can incorporate the scheduling of neutralization sorties within our scheduling algorithm. However, in order to incorporate the physical constraints of allowing a USV to travel to a location, and requiring the transit for a neutralization action, we must include the following constraints. First, we require that in order for a USV to neutralize a target, it must have previously traveled to that target, represented by the constraint

$$\begin{aligned}
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P} \text{ s.t. } p \neq 1, \forall d_n \in \mathcal{D} \\
& \bar{\mathbf{I}}_{n,p,d_n}^{neut} = 1 \Rightarrow \sum_{d'_n \in \mathcal{D}_n \text{ s.t. } \langle d'_n, d_n \rangle \in \mathcal{V}_n} \bar{\mathbf{I}}_{n,p-1,d'}^{move_n} = 1.
\end{aligned} \tag{38}$$

Secondly, in order to allow transits between dock/deploy nodes \mathcal{D} and neutralization nodes \mathcal{D}_n , we require the constraint

$$\begin{aligned}
& \forall n \in \mathcal{A}_s, \forall p \in \mathcal{P} \text{ s.t. } p \neq 1, \forall d_n \in \mathcal{D}_n \\
& \sum_{d'_n \in \mathcal{D}_n \text{ s.t. } \langle d'_n, d_n \rangle \in \mathcal{V}_n} \bar{\mathbf{I}}_{n,p,d_n}^{move_n} = 1 \Rightarrow \\
& \sum_{d'_n \in \mathcal{D}_n \text{ s.t. } \langle d'_n, d_n \rangle \in \mathcal{V}_n} \bar{\mathbf{I}}_{n,p-1,d'}^{move_n} = 1.
\end{aligned} \tag{39}$$

We now conclude the constraints that implement neutralization actions, and turn to the overall cost function used in the optimization process.

4 Complexity and Complexity Reduction

Now that we have fully modeled the MCM scheduling problem, we now discuss initial methods to limit the search space required by MILP solvers. The full optimization is NP hard, which can be shown with the following proofs based on the formal SATP framework.

Proposition 4.1. *The SATP belongs to the class NP.*

Proof. We first pose the corresponding decision problem to the optimization problem constructed from

constraints (2)-(30) and cost function (31) as a formal language.

$$\begin{aligned}
 SATP = \{ & \langle \mathcal{D}, \mathcal{V} \rangle, \mathbf{c}, S, M, N, l, f(\cdot), P, T' \} \\
 & f : \mathcal{D} \rightarrow \mathcal{S} \\
 & T' \in \mathbb{R} \\
 & M \text{ survey UUVs and } N \text{ USVs are} \\
 & \text{assigned tasks during phases } \mathcal{P} = \{1, \dots, P\} \\
 & \text{such that constraints (2)-(30) hold with} \\
 & \max_{p \in \mathcal{P}} \bar{\mathbf{T}}_{a,p}^{end} \text{ of at most } T'.
 \end{aligned} \tag{40}$$

Next, consider a two-input, polynomial time algorithm $A(\cdot)$ that can verify $SATP$. One of the inputs to $A(\cdot)$ is an implementation of constraints (2)-(30). The other input is a binary assignment of the decision variables $\bar{\mathbf{I}}_{a,p,l}^k$, $\bar{\mathbf{T}}_{a,p}^{start}$, $\bar{\mathbf{D}}_{m,p}$, $\bar{\mathbf{F}}_{m,p}$, and $\bar{\mathbf{T}}_{a,p}^{end}$. We construct algorithm A as follows: for each instantiation of the constraints, we check to see if the constraints hold in polynomial time. For implementation of constraints (9), (17)-(30), we first convert them to a standard MILP implementation using techniques such as Big-M [29]. The conversion can also be done in polynomial time. The algorithm then finds the maximum value of $\bar{\mathbf{T}}_{a,p}^{end}$, and checks to see if it is at most T' . This process certainly can be done in polynomial time. \square

Proposition 4.2. *The SATP is NP-complete.*

Proof. To prove the SAT is NP-Complete, we must show that it 1) belongs to NP, and 2) is polynomial-time reducible to another problem that is NP-Complete [30]. Proposition 4.1 satisfies the first criterion. For the second criterion, we will use the Traveling Salesman Problem (TSP), which has been shown to be NP-Complete [30]. We will now show that $SATP$ is polynomial time-reducible to TSP , written $SATP \leq_P TSP$. Let

$$\begin{aligned}
 TSP = \{ & \langle \mathcal{D}_t, \mathcal{V}_t \rangle, \mathbf{c}_t, T' \} \\
 & \mathbf{c}_t : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R} \\
 & T' \in \mathbb{R} \\
 & \langle \mathcal{D}_t, \mathcal{V}_t \rangle \text{ has a tour of at most } T_t.
 \end{aligned} \tag{41}$$

We first construct an algorithm $f_t : TSP \rightarrow SATP$ as follows. Since we make no restriction on node $d \in \mathcal{D}_S$ being exclusive to survey area $s \in \mathcal{S}$ within SATP we assign the edges of \mathcal{D}_t to map one-to-one to individual survey areas, all survey areas reachable by UUVs and transport agents by all edges, and one of the survey areas to serve as the origin node o . Thus, there are $S = |\mathcal{D}_t|$ survey areas in the SATP equivalent. The standard TSP problem assumes a route for only one agent. We therefore restrict all mappings of TSP to the case where $N = M = 1$ within the new SATP problem. Likewise, we may consider the 'phases' within the TSP as equal to twice the number of vertices (one each for transit and survey), plus two for docking and deployment of the single survey agent by the single USV. Thus, we have $P = 2|\mathcal{D}_t| + 2$. For clarity, we will remove the now superfluous subscripts m and n . Let $c^{dock} = c^{deploy} = c^{survey} = 0$ in the new SATP. Let $f_s^{survey} = f_d^{dock} = f_d^{deploy} = f_v^{move} = c^{charge} = 0$. Let $c_{ij}^{move} = c_{ij}^{trans}$ for all $\langle i, j \rangle = v \in \mathcal{V}$. These assignments can be done in polynomial time.

We now show that TSP has a solution of at most T' if and only if the mapped SATP has a solution of at most T' as well. Suppose TSP has a solution of at most T' . The assignment of fuel costs to zero causes constraints (8)-(10) to become trivial, and the combination of $c^{dock} = c^{deploy} = 0$ and $c_{ij}^{move} = c_{ij}^{trans}$ cause there to be no difference in servicing time between a UUV moving or being transported and deployed between survey areas via (16). The assignment $P = 2|\mathcal{D}_t| + 2$ cause the only feasible schedules to be those involving an initial deployment of the single UUV, followed by a TSP tour of alternating UUV transits and servicing actions followed by a single docking, due to constraints (2)-(7) and (30). However, the equal transport/move cost and zero dock and deploy cost assignment causes (31) to degenerate into the simple tour cost of TSP . Thus, the resulting SATP has a minimum time of at most T' as well. Next, suppose that SATP has a completion time of at most T' . Then, since each survey node must have been visited exactly

once due to the choice of P , and all costs are zero with the exception of transit costs, the corresponding TSP problem has a tour cost of at most T' as well. \square

We now present two strategies for increasing the tractability of the SATP and thereby MCM scheduling problem by limiting the required number of variables needed in a particular SATP implementation. The first strategy reduces the number of candidate dock, deploy, and transit nodes to the closest S nodes that provide one node per survey area. The second strategy maintains the number of candidate nodes, but eliminates all transition edges between nodes within the same survey area. Since the embedded routing problem significantly increases the complexity of scheduling problem as a whole, the general idea of both reduction strategies is to decrease the routing problem in intelligent ways.

4.1 Node Reduction Strategy

We first attempt to reduce the number of variables by selecting only a subset of dock and deploy nodes for consideration in the overall optimization framework, specifically the S nodes in a cluster that minimize the distance traveled to visit all nodes in the subset while providing a dock and deploy point to every survey area \mathcal{S} . Formally, we can write this node subset as

$$\mathcal{D}' = \left\{ \underset{\mathcal{D}^*}{\operatorname{argmin}} \sum_{d \in \mathcal{D}^*} c_{ij}^{move} \mid i, j \in \mathcal{D}^*, \mathcal{D} \subset \mathcal{D}, \forall s \in \mathcal{S}, \right. \\ \left. |\mathcal{D}_s^*| = 1 \right\}, \quad (42)$$

where \mathcal{D}_s^* is the number of nodes connected to survey areas within the node subset \mathcal{D}^* . Equation (42) represents a form of the generalized traveling salesman problem (GTSP) [31, 32]. While the GTSP is another NP-hard problem, its use increases the tractability of the SATP in two ways. First, because the GTSP is partly encapsulated within the SATP when determining routes for the individual agents, solving the GTSP before the main schedule optimization will eliminate orders of magnitude more permutations required in the overall optimization than required to solve the GTSP. Second, there are several heuristics that can quickly provide efficient but sub-optimal solutions to the GTSP [31, 32, 33].

4.2 Edge Reduction Strategy

A second candidate strategy for reducing the scale of the SATP is to reduce the number of decision variables by reducing the total number of candidate edges within the problem. Like the node reduction strategy, this is a balancing act between reducing the scale of the problem and eliminating options for the vehicles to travel in order to transit in an efficient manner. An appropriate compromise between the two conflicting needs is in the elimination of intra-survey area edges. Specifically, we create a new edge map \mathcal{V}' as

$$\mathcal{V}' = \{(i, j) \in \mathcal{V} \mid i \in \mathcal{D}_{s_i}, j \in \mathcal{D}_{s_j}, s_i \neq s_j\}. \quad (43)$$

The intuition behind using (43) is the assumption that there is little need for either a survey UUV or USV to transit between two nodes in a single survey area, as once an area is surveyed, the vehicles can simply choose the best route to the next survey area.

Transit paths between four survey areas consisting of four dock and deploy nodes per area are shown in Figure 2. The full graph $\langle \mathcal{D}, \mathcal{V} \rangle$ is shown on the left-hand side of the figure. The graph $\langle \mathcal{D}, \mathcal{V}' \rangle$ resulting from edge reduction strategy is shown in the center. The graph $\langle \mathcal{D}', \mathcal{V} \rangle$ depicting the node reduction strategy is shown on the right. The edge reduction strategy eliminates approximately 18% of the edges in the search space for agent movement. The node reduction dramatically decreases the number of edges and nodes, reducing the number of edges to 8% of the full graph.

5 Robust Scheduling Extensions

We now extend our framework to incorporate uncertainty in the duration of various tasks being performed by the individual vehicles. Our methodology is based on the general idea that there is a significant potential cost

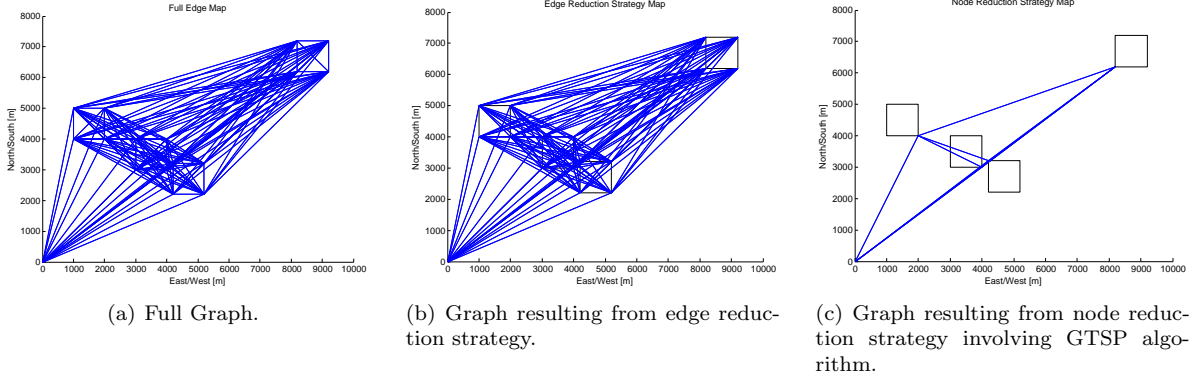


Figure 2: Comparison between full edge transit graph and graph from complexity reduction strategies. Blue represents edge in transit graph \mathcal{V} . Black represents survey area boundaries. Left: Complete graph from all dock/deploy points. Center: Graph from edge reduction strategy. Right: Graph from node reduction strategy using the GTSP algorithm.

of schedule slips. Specifically, we define a *schedule slip* as when the end time of a task as scheduled exceeds the scheduled start time of the following task. Often, schedule slips can have catastrophic consequences in real-world multi-agent systems. For example, an UUV finishing a survey task on time while a USV is significantly delayed could loiter in a dangerous location or potentially run out of fuel before docking could be performed. Similarly, a survey UUV being significantly delayed when expected to rendezvous with a USV could significantly delay the entire schedule for the remainder of the mission. In order to develop our robust scheduling scheme to mitigate these hazards with our robust survey UUV transport problem (R-SATP) extension, we first make the assumption that for tasks taken by the individual vehicles, no tasks can be performed before their scheduled start time. Formally, this can be done by creating a constraint defining the true initial start time of a task as

$$\tilde{\mathbf{T}}_{a,p}^{start} = \max \left\{ \tilde{\mathbf{T}}_{a,p-1}^{end}, \bar{\mathbf{T}}_{a,p}^{start} \right\}, \quad (44)$$

where $\tilde{\mathbf{T}}_{a,p-1}^{end}$ is the true start time of the previous phase for vehicle a . We then define the true start time of the previous phase as

$$\tilde{\mathbf{T}}_{a,p-1}^{end} = \tilde{\mathbf{T}}_{a,p}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_{\parallel}} \tilde{c}_{a,p,l}^k \bar{\mathbf{I}}_{a,p,l}^k, \quad (45)$$

where $\tilde{c}_{a,p,l}^k$ is the true duration of task k during phase p performed by vehicle a on location set l . We now constrain $\tilde{\mathbf{T}}_{a,p+1}^{start}$ from (15) to

$$\tilde{\mathbf{T}}_{a,p+1}^{start} = \bar{\mathbf{T}}_{a,p}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} c_{a,l}^k \bar{\mathbf{I}}_{a,p,l}^k + T_{a,p}^{buff}, \quad (46)$$

where $T_{a,p}^{buff}$ is a notional buffer time to prevent schedule slips that we will define with optimization variables subsequently. Beforehand, we will characterize our general strategy of focusing on minimizing schedule slips during individual phases. We do so with the following proposition, which shows that if a phase is currently in a schedule slip situation, there must have been a phase in the past where the a schedule slip did not occur.

Proposition 5.1. *If*

$$\tilde{\mathbf{T}}_{a,p}^{start} = \tilde{\mathbf{T}}_{a,p-1}^{end} \quad (47)$$

and

$$\tilde{\mathbf{T}}_{a,p}^{start} > \bar{\mathbf{T}}_{a,p-1}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p,l}^k \bar{\mathbf{I}}_{a,p,l}^k, \quad (48)$$

then there exists a $p' \leq p/p = 0$ such that

$$\tilde{\mathbf{T}}_{a,p'}^{start} = \bar{\mathbf{T}}_{a,p'-1}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p,l} \bar{\mathbf{I}}_{a,p,l}^k \quad (49)$$

Proof. Suppose (47) and (48) are true for some p . Since

$$\tilde{\mathbf{T}}_{a,p}^{start} = \tilde{\mathbf{T}}_{a,p-1}^{end} \quad (50)$$

$$= \bar{\mathbf{T}}_{a,p-1}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p,l} \bar{\mathbf{I}}_{a,p,l}^k, \quad (51)$$

then

$$\tilde{\mathbf{T}}_{a,p-1}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p,l} \bar{\mathbf{I}}_{a,p,l}^k > \bar{\mathbf{T}}_{a,p-1}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p,l} \bar{\mathbf{I}}_{a,p,l}^k \quad (52)$$

$$\tilde{\mathbf{T}}_{a,p-1}^{start} > \bar{\mathbf{T}}_{a,p-1}^{start}. \quad (53)$$

Due to the definition of $\tilde{\mathbf{T}}_{a,p}^{start}$ and (52),

$$\tilde{\mathbf{T}}_{a,p-1}^{start} = \tilde{\mathbf{T}}_{a,p-2}^{end} \quad (54)$$

$$= \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k. \quad (55)$$

There are now three possibilities to the relation of $\tilde{\mathbf{T}}_{a,p-1}^{start}$ to

$$\bar{\mathbf{T}}_{a,p-1}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k:$$

$$\tilde{\mathbf{T}}_{a,p-1}^{start} > \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k \quad (56)$$

$$\tilde{\mathbf{T}}_{a,p-1}^{start} < \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k \quad (57)$$

$$\tilde{\mathbf{T}}_{a,p-1}^{start} = \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k. \quad (58)$$

Note that if (58) holds, then our proposition would be proven. Suppose (57) holds. Then

$$\tilde{\mathbf{T}}_{a,p-1}^{start} < \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k \quad (59)$$

$$\tilde{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p,l} \bar{\mathbf{I}}_{a,p-2,l}^k < \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k \quad (60)$$

$$\tilde{\mathbf{T}}_{a,p-2}^{start} < \bar{\mathbf{T}}_{a,p-2}^{start}, \quad (61)$$

which violates our definition of $\tilde{\mathbf{T}}_{a,p}^{start}$ in (44). So we must assume that

$$\tilde{\mathbf{T}}_{a,p-1}^{start} > \bar{\mathbf{T}}_{a,p-2}^{start} + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,p-2,l} \bar{\mathbf{I}}_{a,p-2,l}^k. \quad (62)$$

We now have the same assumptions as before for $p - 1$. By induction, we can follow the same process to $p = 1$. From initial conditions,

$$\tilde{\mathbf{T}}_{a,0}^{start} = \tilde{\mathbf{T}}_{a,0}^{start} = 0 \quad (63)$$

$$\tilde{\mathbf{T}}_{a,1}^{start} = 0 + \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} \tilde{c}_{a,0,l} \bar{\mathbf{I}}_{a,0,l}^k, \quad (64)$$

satisfying the proposition for $p = 1$. \square

We use Proposition 5.1 as motivation to focus on treating the robustification of schedules as a Markov process. Namely, that the dominant factor in the probability of schedule slip in the current phase is the uncertainty in the action performed in the previous phase. In other words, we may mitigate the possibility of a schedule slip in the overall schedule by reducing the possibility of a schedule slip in every phase. Using (44)-(46), we now characterize our general strategy for reducing the risk of schedule slips.

We assume there is a numerical cost (measured in time) to the occurrence of either a schedule slip, or a delay beyond the buffer time

$$\begin{aligned} \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} c_{a,l}^k \bar{\mathbf{I}}_{a,p,l}^k > T^{buff} &\Rightarrow c^{slip} \\ \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}_k} c_{a,l}^k \bar{\mathbf{I}}_{a,p,l}^k < T^{buff} &\Rightarrow c^{ineff} \end{aligned} \quad (65)$$

Using the definition of Bayes risk, see [34], we may write the Bayes risk for this cost definition as

$$R = c^{ineff} P\left(\tilde{c}_{a,l}^k < T_{a,l}^{k,buff}\right) P^{ineff} + c^{slip} P\left(\tilde{c}_{a,l}^k > T_{a,l}^{k,buff}\right) P^{slip}. \quad (66)$$

for a particular action k , where P^{ineff} and P^{slip} are prior probabilities of adverse consequences of either a delay, or a schedule slip, respectively. Varying the buffer time and assuming the task duration follows a $\tilde{c}_{a,l}^k \propto \mathcal{A}_s(0, 20)$ distribution yields a risk function related to buffer time as shown in Figure 3. Note that

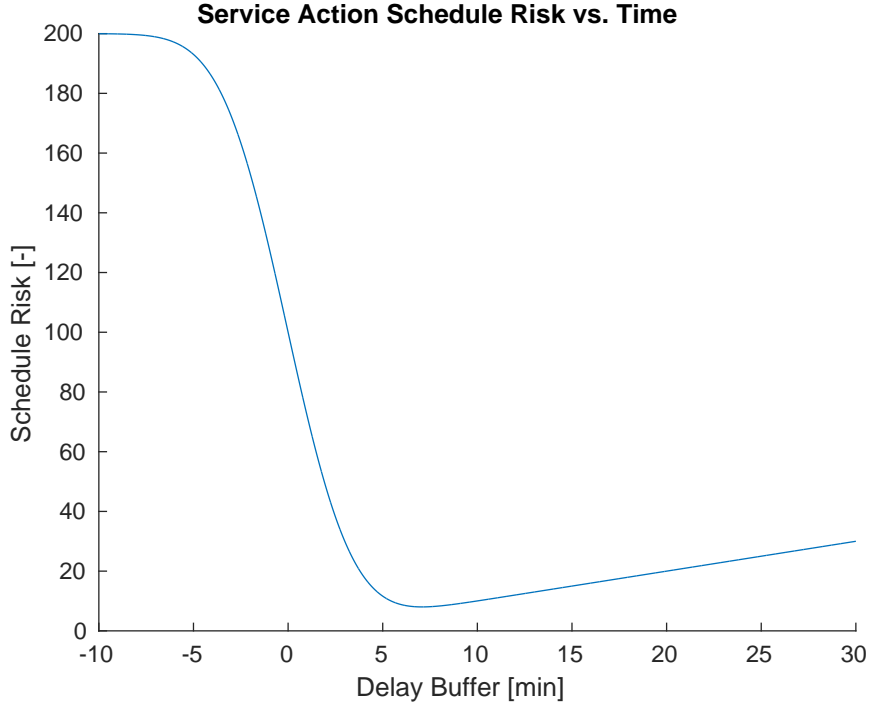


Figure 3: Risk function for Schedule slip based on $\mathcal{A}_s(0, 20)$ with $c^{ineff} = T_{a,l}^{k,buff}$ and $c^{slip} = 200$.

as written, the risk function is invariant in being performed during a particular phase p . As such, we may optimize the nonlinear optimization problem

$$T_{a,l}^{k,buff*} = \underset{T_{a,l}^{k,buff}}{\operatorname{argmin}} c^{ineff} P\left(\tilde{c}_{a,l}^k < T_{a,l}^{k,buff}\right) P^{ineff} + c^{slip} P\left(\tilde{c}_{a,l}^k > T_{a,l}^{k,buff}\right) P^{slip} \quad (67)$$

using numerical techniques prior to the schedule optimization. Once we develop an optimal buffer time for every action, location, and agent (type), we treat them as parameters in our robust scheduling extension. If P^{ineff} and P^{slip} are treated as uniform priors, (67) may be re-written as

$$T_{a,l}^{k,buff*} = \underset{T_{a,l}^{k,buff}}{\operatorname{argmin}} c_{a,l}^{ineff} P\left(\tilde{c}_{a,l}^k < T_{a,l}^{k,buff}\right) + c_{a,l}^{slip} P\left(\tilde{c}_{a,l}^k > T_{a,l}^{k,buff}\right). \quad (68)$$

Once the optimal buffer times for each individual task are found using (67), they may be factored into the overall optimization process. In order to incorporate the buffer time, we take advantage of the necessary delays between coordinated agent actions in the MCM scheduling problem by using the wait times as part of the required buffer times. We define the synchronization time for agent a as the time difference between the end time of the current phase and the proceeding phase, written

$$\bar{\mathbf{T}}_{a,p}^{sync} = \bar{\mathbf{T}}_{a,p+1}^{start} - \bar{\mathbf{T}}_{a,p}^{end}. \quad (69)$$

We then ensure that the decision variable representing the *scheduled* buffer time $\bar{\mathbf{T}}_{a,p}^{sync}$ is at least the desired, optimal buffer time $T_{a,l}^{k,buff}$ with the constraint

$$\bar{\mathbf{I}}_{a,p,l}^k = 1 \wedge \bar{\mathbf{T}}_{a,p}^{sync} \leq T_{a,l}^{k,buff} \Rightarrow \bar{\mathbf{T}}_{a,p}^{buff} = T_{a,l}^{k,buff} - \bar{\mathbf{T}}_{a,p}^{sync}. \quad (70)$$

If the synchronization time is greater than the optimal buffer time parameter $T_{a,l}^{k,buff}$, then the buffer time is set to zero with the constraint

$$\bar{\mathbf{I}}_{a,p,l}^k = 1 \wedge \bar{\mathbf{T}}_{a,p}^{sync} \geq T_{a,l}^{k,buff} \Rightarrow \bar{\mathbf{T}}_{a,p}^{buff} = 0 \quad (71)$$

Finally, the robust start time is set as the deterministic start time plus the summation of all buffer times with the constraint

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P} \quad (72)$$

$$\bar{\mathbf{T}}_{a,p+1}^{start,R} = \bar{\mathbf{T}}_{a,p+1}^{start} + \sum_{p' \in \mathcal{P} \text{ s.t. } p' \leq p} \bar{\mathbf{T}}_{a,p'}^{buff}.$$

In order to complete our robust scheduling extension, we modify (16) and (17) to use the new robust start times, creating the constraints

$$\forall a \in \mathcal{A}, \forall p \in \mathcal{P} \quad (73)$$

$$\bar{\mathbf{T}}_{a,p}^{end} = \bar{\mathbf{T}}_{a,p}^{start} + \sum_{k \in \mathcal{K}} \sum_{\mathcal{L} \in \mathcal{L}_k} c_{a,l}^k \bar{\mathbf{I}}_{a,p,l}^k$$

$$\forall m \in \mathcal{A}_{ur}, \forall m \in \mathcal{A}_s, \forall p \in \mathcal{P} \quad (74)$$

$$\sum_{d \in \mathcal{D}} \bar{\mathbf{I}}_{mn,p,d}^{dock} = 1 \Rightarrow \bar{\mathbf{T}}_{m,p}^{start} = \bar{\mathbf{T}}_{n,p}^{start}$$

$$\forall m \in \mathcal{A}_{ur}, \forall m \in \mathcal{A}_s, \forall p \in \mathcal{P} \quad (75)$$

$$\sum_{d \in \mathcal{D}} \bar{\mathbf{I}}_{mn,p,d}^{deploy} = 1 \Rightarrow \bar{\mathbf{T}}_{m,p}^{end} = \bar{\mathbf{T}}_{n,p}^{end}.$$

6 Simulation Results

The following simulations demonstrate the efficacy and computational performance of the proposed scheduling framework. All simulations were performed using the IBM's ILOG CPLEX optimization software [35]. All simulations were performed on a 2.8 GHz quad-core CPU equipped with 8 GB RAM.

6.1 End-to-End MCM Mission Scheduling

We first demonstrate an end-to-end MCM mission optimization simulation, where the scheduler automatically determines a near-optimal schedule for performing the survey, reacquire and identify, and neutralization components of the MCM mission. Figures 4 and 5 illustrate a simulation of $N = 1$ USV, $M = 2$ survey UUVs, $R = 2$ RI UUVs, and $P = 26$ phases. In the simulations, we assume that a USV has a speed of 8 m/s, while UUVs have a speed of 1.5 m/s. We assume each of the $S = 4$ survey areas require $c^{survey} = 300$ minutes. RI and Neutralize tasks require $c^{RI} = 150$ and $c^{Neut} = 60$ minutes, respectively. Furthermore, we assume that each docking action requires $c^{dock} = 60$ minutes, and $c^{deploy} = 30$ minutes. Post-mission analysis time for the survey and neutralize phases required $\hat{c}_s^{PMA'} = 240$ and $\hat{c}_{RI}^{PMA'} = 15$ minutes, respectively. Figure 4 represents the spatial map of the various vehicles moving throughout the field. Figure 5 shows a temporal schedule of the overall mission. The MILP optimization software was run with a cutoff time of 10 minutes for each optimization phase (Survey-RI, and RI-Neutralize). For the purposes of this simulation, we assumed that all targets were identified as mines, and therefore no additional Neutralize-only phase was necessary. As seen in the figure, the USV travels to the center-most survey area, survey area 2, to deploy the survey vehicles. The USV then travels to a further area to deploy the RI vehicles. The USV then commences to travel about the area docking with various vehicles as the vehicle travel under their own power to maximize the speed at which post-mission analysis commences, including docking with a vehicle to immediately deploy it again in order to survey the final area. It is noted that the UUVs largely use their own motion in order to actually travel to areas to prosecute. We suspect this is due to sacrificing speed on-station for maximizing PMA processing speed.

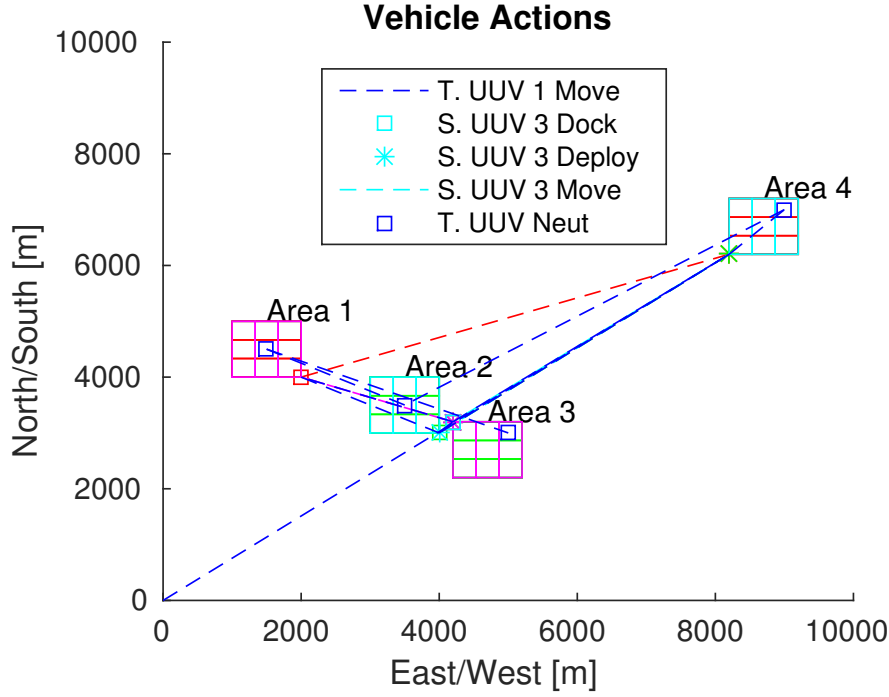


Figure 4: Plot of vehicle actions within an MCM area.

6.2 Deterministic and Robust Scheduling Results

We now present a comparison between the deterministic scheduling of a single phase of an MCM mission, the survey phase, the proposed robust scheduling approach, and a naive robust scheduling approach serving as a baseline. In the following simulations, we assume that the USV has a speed of 8 m/s, while UUVs have a speed of 1.5 m/s. We assume each of the $S = 4$ survey areas requires $c^{survey} = 100$ minutes. Furthermore, we assume that each docking action requires $c^{dock} = 20$ minutes, and $c^{deploy} = 10$ minutes. For the task

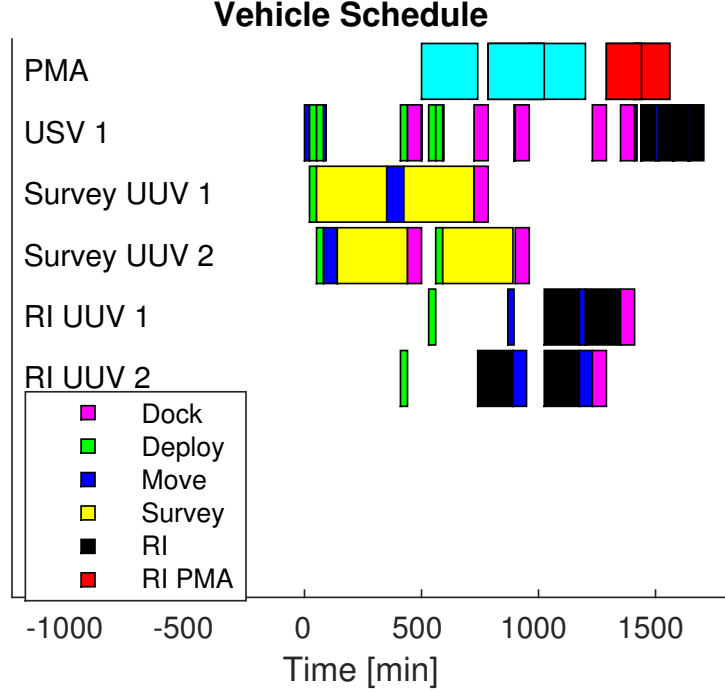


Figure 5: Simulated MCM Schedule. Gantt chart represents the duration of each action performed by each vehicle at a point in time. Additionally, PMA processing times are shown at the top. Blue PMA times represents PMA for survey components of mission. Red PMA times represent RI components. Green and magenta represent vehicle deployment and docking, respectively. Blue represents vehicle transits. Yellow represents surveys. Black on RI vehicles represents RI actions. Black on the USV row represents neutralization actions.

uncertainty distributions, we used $\sigma_{dock}^2 = 10$, $\sigma_{deploy}^2 = 1$, $\sigma_{survey}^2 = 30$, and $\sigma_{move}^2 = \frac{\|<i,j>\|}{10}$ for nodes i and j , respectively. We present a direct comparison of simulations between the deterministic optimization problem, a naive robust optimization problem, and the minimum-risk optimization problem discussed in Section 5 for $N = 1$ USV and $M = 2$ survey agents within $P = 12$ phases. A naive method for mitigating concerns of schedule slips is typically to add a fixed amount of buffer time, often as a function of the standard deviations. As such, to compare the deterministic method to our robust counterpart, we provide a baseline robust simulation where three standard deviations of time required are added for each action type. For example, a survey action could be planned for $c_{survey}^{survey} = 116.425$ seconds. For the minimum-risk optimization problem, we set $c_{a,l}^{slip} = 500$ and $c_{a,l}^{ineff} = T_{a,l}^{buff}$ for all task types.

Figure 6 shows a comparison of a scenario using the three different optimization problems. As expected, the deterministic solution has the best completion time of 310.9 minutes. The naive robust solution had a completion time of 364.9 minutes. The minimum-risk robust scheduling solution had a completion time of 345.45 minutes. Also of note is the different strategies taken by the deterministic or naive solution, and the minimum-risk solution. The deterministic and naive robust solutions both schedule the USV (labeled transport agent) to pick up and then re-deploy one of the UUVs (labeled survey agents) to the two furthest survey areas. The minimum-risk scheduling algorithm has the USVs both deployed at the closest survey areas, and one travels to each of the furthest areas under its own power. We conjecture that this is likely due to the minimum-risk scheduling taking into account the increased schedule slip risk due to executing an increased number of tasks required for dock and re-deployment versus a single move action. We empirically analyze the computational performance of the minimum-risk robust solution versus the deterministic solution in the following section.

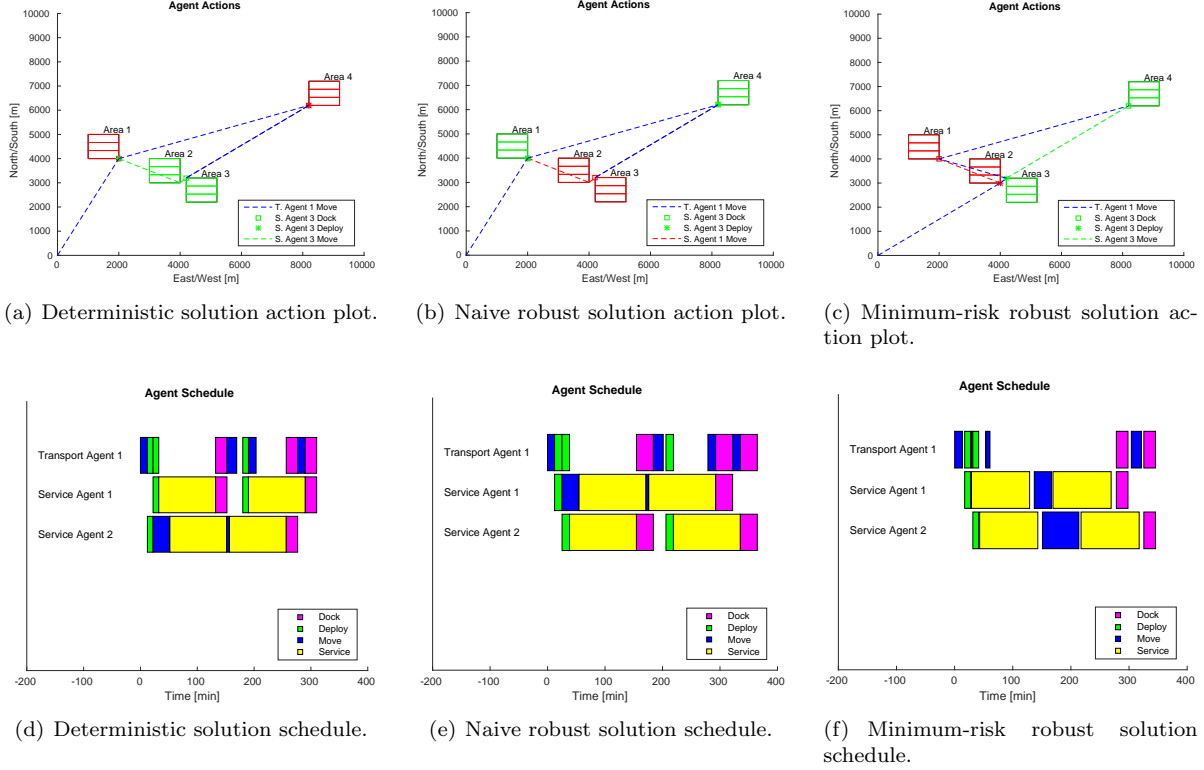


Figure 6: Comparison between deterministic optimization, a naive robust optimization approach, and the proposed minimum-risk scheduling approach.

6.3 Analysis of Complexity Reduction Heuristics and Computational Tractability

Figure 7 shows results from a Monte Carlo simulation comparing the full graph to the two complexity reduction strategies discussed in Section 4 for an SATP optimization problem involving $N = 1$ USV, $M = 3$ survey UUVs, and $P = 15$ phases. Sixty simulations were performed with three survey areas randomly generated over a 10,000m by 10,000m field. For each simulation, the cutoff time for providing the best solution from CPLEX for the optimization problem was 120 seconds. As seen in the figure, the node reduction strategy resulted in the fastest sortie time, averaging an objective value (31) of 281.4 minutes. The edge-reduction strategy provided a modest improvement over the full optimization model, averaging an objective value of 373.1. As expected, the full optimization model, due to its significantly greater complexity, resulted in the poorest solutions, averaging 398.8 minutes. Additionally, in some instances, the optimization solver was unable to find a solution to the problem in the specified time limit due to the computational complexity of the full optimization problem. The timeout without a solution occurred in 48% of the full optimization problem's simulations, and 23% of simulations using the edge reduction strategy. However, in no instances of using the node reduction strategy did a timeout occur.

A second Monte Carlo simulation was performed involving sixty simulations of a single USV and survey UUV, and $P = 12$ phases, allowing all optimization routines to run until a globally optimal solution is found. In this manner, the Monte Carlo simulation characterized how the two heuristic reduction strategies (42) and (43) impact the discovery of the globally optimal schedule by eliminating transit paths. In all cases, the globally optimal solution to the full optimization problem was identical to the globally optimal solution of the edge reduction strategy (43). The node reduction strategy (42) did eliminate the globally optimal solution in all instances, however this elimination increased the minimum solution's objective function value by an average of only 3.45%, and maximum of 7.7%.

In addition to the simulations characterizing the effects of the various heuristics on the quality of the opti-

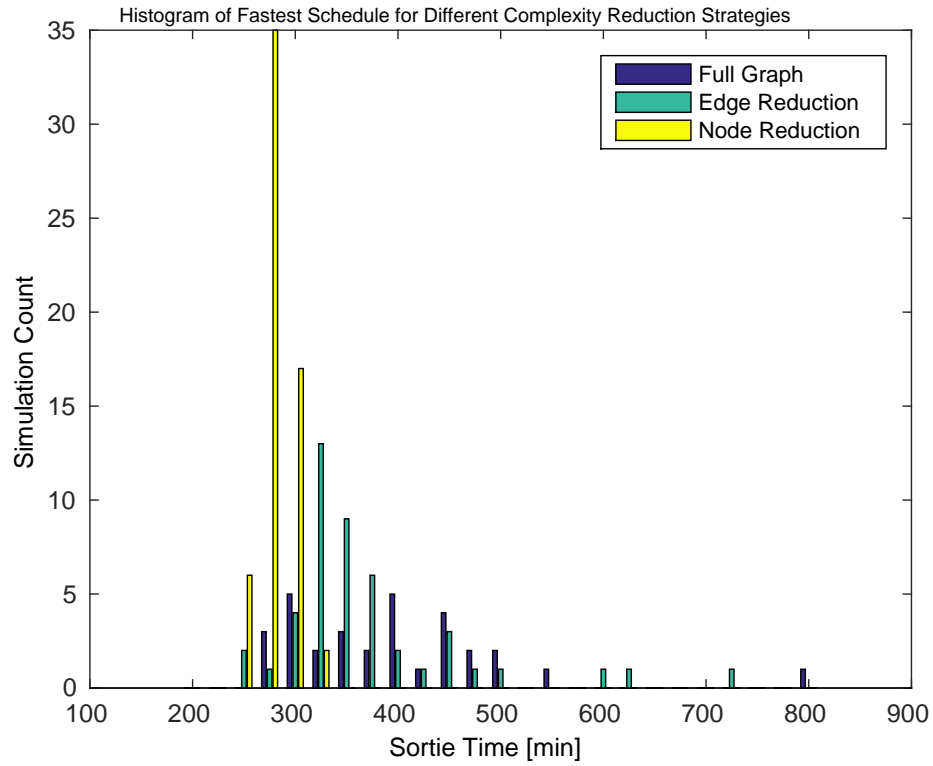


Figure 7: Histogram of Monte Carlo simulations showing minimum objective value found by different complexity reduction strategies.

Table 1: Characterization of phase, agent count, and optimization type effects on optimization quality. Times represent the schedule’s objective value in minutes after 5 minutes optimization runtime.

Agent Count	Opt. Type	# of Phases				
		12	14	16	18	20
2	Standard	522.86*	507.72*	507.72*	507.72*	507.72*
	Robust	567.61	559.12	567.61	567.61	567.61
3	Standard	297.72	295.16	303.57	303.57	303.57
	Robust	337.12	338.01	349.68	360.39	384.61
4	Standard	244.14	244.14	244.14	243.55	244.14
	Robust	377.62	314.24	333.70	373.47	390.96
5	Standard	271.01	271.01	281.41	296.43	275.08
	Robust	313.44	513.45	389.08	409.80	277.45
6	Standard	173.02	188.44	236.97	171.41	192.42
	Robust	424.41	379.24	275.37	435.07	—

mization result, we wish to characterize the effect of increasing number of (binary) optimization variables on the optimization framework while using the node reduction heuristic. Table 1 shows a number of simulations that vary the agent count and number of phases in which to schedule tasks. Additionally, the simulations show the difference between the robust versus deterministic SATP case, and the result of the optimization yielding the best solution from CPLEX after 300 seconds of run-time. If computational tractability were irrelevant, the best solution would be found from the problem formulation involving the most phases and agents, since the scheduler could take advantage of tasking multiple agents, and slotting those agents in complicated task schedules. Asterisks represent where the globally optimal schedule was found within the cutoff time. For $A \geq 5$ agents, two USVs were used in the scenario. As seen in the table, the lower number of phases/agents resulted in a higher likelihood of the globally optimal schedule being found. As a larger number of agents and phases were used, the results were generally sub-optimal, and at times a worse result than a fewer number of phases for the same agent number, due to the dramatically increased search space due to the binary variables. From the table and judging by the variation in best search schedules returned before the cut-off time, reasonable numbers of agents for the SATP scheduling problem as developed appears to be four agents or fewer for the deterministic schedule, and three agents and fewer than $P = 14$ phases for the robust scheduling extension. The only scenario that failed to produce a result was the most complicated of six agents and 20 phases using the robust extension, which created an optimization problem of 26,190 binary variables. This compares to the simplest scenario, which created an optimization problem of 869 binary variables.

7 Conclusions

We have presented a framework for optimizing scheduling of multiple heterogeneous vehicles in order to perform an MCM survey mission. The framework utilizes mixed-integer linear programming in order to obtain optimal to near-optimal solutions, depending on the complexity of the problem and length of time the algorithm is allowed to search for a result. Future work includes extending the framework to re-acquire and identify tasks, as well as neutralization tasks. Additionally, adding probabilistic constraints in order to account for uncertainty in the length of time required will be explored. Lastly, we hope to explore the use of pre-processing heuristics to reduce the size and complexity of the MILP problem and lead to faster solve times for efficient solutions. The framework utilizes mixed-integer linear programming techniques to create novel constraints required to develop a tasking schedule for the vehicles including docking, deployment, and movement actions performed throughout the area. A robust scheduling technique was introduced that extends the framework to include probabilistic constraints in order to account for uncertainty in the length of time required to perform tasks, as well as exploring decentralized optimization strategies to increase the scalability of the optimization framework. Future work includes developing a decentralized optimization process to make the scheduling algorithms more computationally tractable for larger numbers of vehicles.

References

- [1] A. J. Shafer, M. R. Benjamin, J. J. Leonard, J. Curcio, Autonomous cooperation of heterogeneous platforms for sea-based search tasks, in: MTS/IEEE OCEANS, 2008.
- [2] E. G. Jones, B. Browning, M. B. Dias, B. Argall, M. M. Veloso, Dynamically formed heterogeneous robot teams performing tightly coordinated tasks, in: IEEE International Conference on Robotics and Automation, 2006.
- [3] T. Estlin, D. Gaines, F. Fisher, R. Castano, Coordinating multiple rovers with interdependent science objectives, in: Proceedings of the fourth international joint conference on autonomous agents and multiagent systems, 2005.
- [4] P. Sujit, A. Healey, J. B. Sousa, AUV docking on a moving submarine using a K-R navigation function, in: International Conference on Intelligent Robots and Systems, 2011.
- [5] B. W. Hobson, R. S. McEwen, J. Erickson, T. Hoover, L. McBride, F. Shane, J. G. Bellingham, The development and ocean testing of an AUV docking station for a 21" AUV, in: IEEE/MTS OCEANS, 2007.
- [6] A. Martins, J. M. Almeida, H. Ferreira, H. Silva, N. Dias, A. Dias, C. Almeida, E. Silva, Autonomous surface vehicle docking manoeuvre with visual information, in: International Conference on Robotics and Automation, 2007.
- [7] J. N. Weaver, D. Z. Frank, E. M. Swartz, A. A. Arroyo, UAV performing autonomous landing on USV utilizing the robot operating system, in: ASME Early Career Technical Symposium, 2013.
- [8] K. E. Wenzel, A. Masselli, A. Zell, Automatic takeoff, tracking and landing of a miniature UAV on a moving carrier vehicle, *Journal of Intelligent Robot Systems* 61 (2011) 221–238.
- [9] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*, Oxford University Press, 1995.
- [10] M. J. Bays, A. Shende, D. J. Stilwell, An approach to multi-agent area protection using Bayes risk, in: International Conference on Robotics and Automation, 2012.
- [11] M. Koes, I. Nourbakhsh, K. Sycara, Constraint optimization coordination architecture for search and rescue robotics, in: International Conference on Robotics and Automation, 2006.
- [12] G. A. Korsah, A. Stentz, M. B. Dias, I. Fanaswala, Optimal vehicle routing and scheduling with precedence constraints and location choice, in: International Conference on Robotics and Automation, 2010.
- [13] C. Mandalandraki, M. S. Daskin, Time dependent vehicle routing problems: Formulations, properties, and heuristic algorithms, *Transportation Science* 26 (1992) 185–200.
- [14] N. Mathew, S. L. Smith, S. L. Waslander, Optimal path planning in cooperative heterogeneous multi-robot delivery systems, in: Workshop on the Algorithmic Foundations of Robotics, 2014.
- [15] M. C. Gombolay, R. J. Wilcox, J. A. Shah, Fast scheduling of multi-robot teams with temporospatial constraints, in: *Robotics: Science and Systems*, 2013.
- [16] R. Dechter, Temporal constraint networks, *Artificial Intelligence* 49 (1991) 61–95.
- [17] S. Sariel, T. Balch, N. Erdogan, Naval mine countermeasure missions, *IEEE Robotics & Automation Magazine* 15 (2008) 45–52.
- [18] R. Costa, T. A. Wettergren, Managing the areas of responsibility for collaborative undersea searchers, in: IEEE/MTS OCEANS, 2010.
- [19] J. C. Hyland, C. M. Smith, Automated area segmentation for ocean bottom surveys, in: *Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, 2015.

- [20] M. Molineaux, B. Auslander, P. G. Moore, K. M. Gupta, Minimally disruptive schedule repair for mcm missions, in: Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX, 2015.
- [21] X. Lin, S. L. Janak, C. Floudas, A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty, Computers & Chemical Engineering 29 (2003) 1069–1085.
- [22] S. L. Janak, X. Lin, C. Floudas, A new robust optimization approach for scheduling under uncertainty: Ii. uncertainty with known probability distribution, Computers & Chemical Engineering 31 (2006) 171–195.
- [23] D. Bertsimas, M. Sim, The price of robustness, Operations Research 52 (1) (2004) 35–53.
- [24] R. L. Daniels, P. Kouvelis, Robust scheduling to hedge against processing time uncertainty in single-stage production, Management Science 41 (2) (1995) 363–376.
- [25] M. Lombardi, M. Milano, L. Benini, Robust scheduling of task graphs under execution of time uncertainty, IEEE Transactions on Computers.
- [26] A. J. Davenport, C. Gefflot, J. C. Beck, Slack-based techniques for robust schedules, in: Sixth European Conference on Planning, 2014.
- [27] S. D. Wu, E.-S. Byeon, R. H. Storer, A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness, Operations Research 47 (1) (1999) 113–124.
- [28] M. J. Bays, T. A. Wettergren, A solution to the service agent transport problem, in: IEEE International Conference on Intelligent Robots and Systems, To appear.
- [29] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, Automatica 35 (3) (1999) 407 – 427. doi:DOI: 10.1016/S0005-1098(98)00178-2.
- [30] T. . Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, 2009.
- [31] L. Davis, Applying adaptive algorithms to epistatic domains, in: International Joint Conference on Artificial Intelligence, 1985.
- [32] M. Fichetti, J. J. S. Gonzalez, P. Toth, A branch and cut algorithm for the symmetric generalized traveling salesman problem, Operations Research 45 (1997) 378–394.
- [33] J. Silberholz, B. Golden, The generalized traveling salesman problem: A new genetic algorithm approach, Extended Horizons: Advances in Computing, Optimization, and Decision Technologies 37 (2007) 165–181.
- [34] H. V. Poor, An Introduction to Signal Detection and Estimation, Springer, 1994.
- [35] IBM Corporation, IBM ILOG CPLEX User Manual (2013).

A Distribution

Defense Technical Information Center ATTN DTIC-0 8725 John J. Kingman Road Fort Belvoir, VA 22060-6218	1
Office of Naval Research ATTN: Code 32, One Liberty Center, Suite 1425 875 North Randolph Center, Suite 1425 Arlington, VA 22203-1995	1
Naval Surface Warfare Center, Panama City Division ATTN: Technical Library 110 Vernon Avenue Panama City, FL 32407	1

